



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра автоматики та управління в технічних системах

**Лабораторна робота №5**  
**Архітектура комп'ютера**  
«УПРАВЛІННЯ ХОДОМ ВИКОНАННЯ ПРОГРАМИ НА АСЕМБЛЕРІ В  
АРХІТЕКТУРІ IA-32 (X86) У REAL ADDRESS MODE»

Виконали:  
студенти групи ІТ-01:

Перевірив:  
Бердник Ю. М.

Бардін В.Д.  
Задніпрянець А.А.  
Куркін О.О.

Дата здачі 08.04.2021

Захищено з балом \_\_\_\_\_

Київ 2021

## Вихідний код:

```
1 TITLE ЛР_5
2 ;-----
3 ;ЛР №5
4 ;-----
5 ; Комп'ютерна архітектура
6 ; ВУЗ: НТУУ "КПІ"
7 ; Факультет: ФІОТ
8 ; Курс: 1
9 ; Група: ІТ-01
10 ;-----
11 ; Автор: Бардін В. Д.
12 ; Задніпрянець А. А.
13 ; Куркін О. О.
14 ;
15 ; Дата: 04/04/21
16 ;-----
17 IDEAL ; Директива - тип Асемблера tasm
18 MODEL small ; Директива - тип моделі пам'яті
19 STACK 2048 ; Директива - розмір стеку
20
21 DATASEG
22 array_stack dw 2E2Eh, 2098h, 6847h, 7230h, 5182h, 1964h, 7350h, 1628h, 9823h, 5927h, 2343h, 7515h, 5238h, 8272h, 6576h, 2619h
23 dw 9520h, 6608h, 5535h, 9745h, 8314h, 1683h, 2065h, 3718h, 3198h, 5267h, 1546h, 2273h, 9317h, 2181h, 7466h, 1940h
24 dw 7417h, 2385h, 2240h, 8346h, 9246h, 9573h, 5050h, 1549h, 4402h, 7854h, 8126h, 9060h, 3476h, 7497h, 3703h, 1857h
25 dw 8794h, 8017h, 3227h, 1033h, 7980h, 8658h, 6475h, 2653h, 4970h, 3343h, 3788h, 4600h, 4953h, 5156h, 7128h, 9539h
26 dw 4889h, 8734h, 8685h, 3104h, 5514h, 9721h, 5958h, 4611h, 7759h, 1725h, 6059h, 7499h, 9681h, 4573h, 6929h, 1387h
27 dw 2772h, 5484h, 7392h, 8250h, 5503h, 6080h, 1249h, 3413h, 3167h, 9250h, 7496h, 7533h, 2101h, 4007h, 6810h, 4531h
28 dw 4979h, 9050h, 8589h, 6962h, 5374h, 3451h, 2971h, 6612h, 8002h, 4074h, 2634h, 3694h, 7979h, 7571h, 7333h, 9852h
29 dw 9847h, 3270h, 2426h, 6722h, 2697h, 9765h, 2203h, 3222h, 6819h, 5024h, 5846h, 3619h, 9626h, 2638h, 4465h, 2E2Eh
30
31 first_birthdate db "08102002"
32 second_birthdate db "09102002"
33 third_birthdate db "24042003"
34 default_birthdate db "00000000"
35 CODESEG
36 Start:
37 mov ax, @data ; data segment init
38 mov ds, ax
39 mov es, ax
40
41 ; set params for copy_array
42 mov cx, 128 ; repeats amount
43 call copy_arr
44
45 ; set params for set_bdates
46 mov cx, 8
47 mov bp, 0151h
48 call set_bdatesd
49
50 ; set params for set_bdates
51 mov cx, 8
52 mov bp, 0161h
53 call set_bdates1
54
55 ; set params for set_bdates
56 mov cx, 8
57 mov bp, 0171h
58 call set_bdates2
59
60 ; set params for set_bdates
61 mov cx, 8
62 mov bp, 0181h
63 call set_bdates3
64
65 ; set params for set_bdates
66 mov cx, 8
67 mov bp, 0191h
68 call set_bdatesd
69
70 ; set params for set_bdates
71 mov cx, 8
72 mov bp, 01A1h
73 call set_bdatesd
74
75 ; set params for set_bdates
76 mov cx, 8
77 mov bp, 01B1h
78 call set_bdatesd
79
80 ; set params for set_bdates
81 mov cx, 8
82 mov bp, 01C1h
83 call set_bdatesd
84
85 ; application finishing
86 mov ah, 4ch
87 int 21h
```

```

88 ;-----Copy array procedure-----
89 ; Input params: cx - initial array size,
90 ; Output params: array copy placed at ds,
91 ;-----
92 PROC copy_arr
93     xor si, si ; set si to zero
94     array_coping_loop:
95         mov bx, [ds:si] ; get number from array_array stack & set it to bx as a temp variable
96         mov [ds:[si+260h]], bx ; set value from bx to ds with offset
97         add si, 2 ; si value + 2
98     loop array_coping_loop
99
100     ret
101 ENDP
102
103 ;-----Add birthdate to stack-----
104 ; Input params: cx - symbols amount at birthday,
105 ; bp - offset
106 ; Output params: array copy placed at ds,
107 ;-----
108 PROC set_bdates1
109     xor si, si ; set si to zero
110     birthdate1_label:
111         mov ah, [first_birthdate+si] ; set value to ah from first_birthdate with offset si
112         mov [bp], ah ; add value to stack
113         inc si ; increment si
114         inc bp ; increment bp
115     loop birthdate1_label
116
117     ret
118 ENDP
119
120 ;-----Add birthdate to stack-----
121 ; Input params: cx - symbols amount at birthday,
122 ; bp - offset
123 ; Output params: array copy placed at ds,
124 ;-----
125 PROC set_bdates2
126     xor si, si ; set si to zero
127     birthdate2_label:
128         mov ah, [second_birthdate+si] ; set value to ah from second_birthdate with offset si
129         mov [bp], ah ; add value to stack
130         inc si ; increment si
131         inc bp ; increment bp
132     loop birthdate2_label
133
134     ret
135 ENDP
136
137 ;-----Add birthdate to stack-----
138 ; Input params: cx - symbols amount at birthday,
139 ; bp - offset
140 ; Output params: array copy placed at ds,
141 ;-----
142 PROC set_bdates3
143     xor si, si ; set si to zero
144     birthdate3_label:
145         mov ah, [third_birthdate+si] ; set value to ah from third_birthdate with offset si
146         mov [bp], ah ; add value to stack
147         inc si ; increment si
148         inc bp ; increment bp
149     loop birthdate3_label
150
151     ret
152 ENDP
153
154 ;-----Add birthdate to stack-----
155 ; Input params: cx - symbols amount at birthday,
156 ; bp - offset
157 ; Output params: array copy placed at ds,
158 ;-----
159 PROC set_bdatesd
160     xor si, si ; set si to zero
161     dbirthdate_label:
162         mov ah, [default_birthdate+si] ; set value to ah from third_birthdate with offset si
163         mov [bp], ah ; add value to stack
164         inc si ; increment si
165         inc bp ; increment bp
166     loop dbirthdate_label
167
168     ret
169 ENDP
170
171
172 end Start

```

**Результати дослідження:** Було створено двовимірний масив, що складається з елементів у два байти (слово) кожен, має розмір 16x8

(для зручності виведення). Потім за допомогою процедури `copy_arr` було зроблено копію масиву.

```
;-----Copy array procedure-----
; Input params: cx - initial array size,
; Output params: array copy placed at ds,
;-----
PROC copy_arr
    xor si, si                ; set si to zero
    array_coping_loop:
        mov bx, [ds:si]      ; get number from array & set it to bx
        mov [ds:[si+260h]], bx ; set value from bx to ds with offset
        add si, 2            ; si value + 2
        loop array_coping_loop

    ret
ENDP
```

Далі за допомогою процедури `set_bdates` та декількох її модифікацій було додано в масив область 8x8 байт, яка складається з дат народження учасників команди.

```
;-----Add birthdate to stack-----
; Input params: cx - symbols amount at birthday,
;               bp - offset
; Output params: array copy placed at ds,
;-----
PROC set_bdates
    xor si, si                ; set si to zero
    birthdate_label:
        mov ah, [first_birthdate+si]
        mov [bp], ah         ; add value to stack
        inc si               ; increment si
        inc bp               ; increment bp
        loop birthdate_label

    ret
ENDP
```

```
DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
[ ]=Dump 3 [ ]
ds:0200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ds:0210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ds:0220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ds:0230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ds:0240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ds:0250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ds:0260 2E 2E 98 20 47 68 30 72 82 51 64 19 50 73 28 16 .ü Gh0réQd4Ps(
ds:0270 23 30 30 30 30 30 30 30 30 52 72 82 76 65 19 26 #00000000Réve1&
ds:0280 20 30 38 31 30 32 30 30 32 83 83 16 65 20 18 37 08102002âme 17
ds:0290 98 30 39 31 30 32 30 30 32 93 81 21 66 74 40 19 y09102002ü!ft@1
ds:02A0 17 32 34 30 34 32 30 30 33 92 73 95 50 50 49 15 #24042003Is0PPI&
ds:02B0 02 30 30 30 30 30 30 30 30 34 97 74 03 37 57 18 00000000iùt7Wf
ds:02C0 94 30 30 30 30 30 30 30 30 79 58 86 75 64 53 26 00000000jXâudS&
ds:02D0 70 30 30 30 30 30 30 30 30 49 56 51 28 71 39 95 p00000000IUQ(q9d
ds:02E0 89 30 30 30 30 30 30 30 30 55 21 97 58 59 11 46 00000000J!ùXY4F
ds:02F0 59 77 25 17 59 60 99 74 81 96 73 45 29 69 87 13 Ywz:1Y'ûtuûsE)ic!!
ds:0300 72 27 84 54 92 73 50 82 03 55 80 60 49 12 13 34 r'âTfâsPé♥UQ`I!4
ds:0310 67 31 50 92 96 74 33 75 01 21 07 40 10 68 31 45 g1Pffût3u@!•@h1E
ds:0320 79 49 50 90 89 85 62 69 74 53 51 34 71 29 12 66 yIPÉëàbitSQ4q)†f
ds:0330 02 80 74 40 34 26 94 36 79 79 71 75 33 73 52 98 0Çt@4&ö6yyqu3sRij
ds:0340 47 98 70 32 26 24 22 67 97 26 65 97 03 22 22 32 Gÿp2&$"gù&eu♥""2
Ctrl: G-Goto S-Search N-Next C-Change F-Follow P-Previous D-Display B-Block
```

**Висновок:** У ході лабораторної роботи попрацювали з різними видами адресації, створили двовимірний масив і розібралися як працює найшвидший спосіб роботи з даними: стек. А також доопрацювали і адаптували код з лабораторної роботи №4 для задачі з ЛР №5.